RSX–11M/M–PLUS
Error Logging Reference Manual

Order No. AA-H270A-TC

RSX-11M Version 3.2
RSX-11M-PLUS Version 1.0

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, June 1979

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

CONTENTS

CONTENTS

FIGURES

TABLES

PREFACE

MANUAL OBJECTIVES AND INTENDED AUDIENCE

This manual contains information needed to operate the error logging subsystem. It tells the user how to produce error logging reports. These reports allow the user to monitor the hardware reliability of an RSX-11M/M-PLUS system. They also provide DIGITAL field service engineers with a convenient history of system performance.

This manual assumes a familiarity with the following documents:

The RSX-11M/M-PLUS MCR Operations Manual

The RSX-11M Utilities Manual

The RSX-11M-PLUS or RSX-11M System Generation and Management Guide

STRUCTURE OF THIS DOCUMENT

Chapter 1 describes the purposes and functions of error logging.

Chapter 2 describes how the Executive logging features and the error logging tasks interface to produce the final, readable reports.

Chapter 3 describes the procedures for operating the error logging tasks.

Chapter 4 describes the error log reports generated by the subsystem.

Chapter 5 describes the messages generated by error logging tasks.

Appendix A explains how to modify the analyzer task SYE and the error logging task ERL at task build time.

Appendix B describes the record formats of temporary files produced by the error logging task ERRLOG.

Appendix C describes record formats contained in the file ERROR.SYS.

Appendix D discusses modifications made to mass storage device drivers to enable error logging. It also discusses several Executive routines supplied for the error logging subsystem.

ASSOCIATED DOCUMENTS

The RSX-11M and RSX-11M-PLUS documentation directories define the intended readership of each manual in the documentation set and provide a brief synopsis of each manual's contents. References are made in the error logging manual to several other documents; readers should consult the appropriate directory for information about these documents. The RSX-11M and RSX-11M-PLUS Mini-References contain a section summarizing Error Logging Task operations.

CHAPTER 1

INTRODUCTION

1.1 THE PURPOSES OF ERROR LOGGING

The RSX-11M/M-PLUS error logging subsystem monitors the hardware reliability of an RSX-11M/M-PLUS operating system. It continually receives information from the executive about every disk, DECtape, magnetic tape, and memory parity error, regardless of whether or not the error is recoverable. Error logging records these events and at user-determined intervals, a report-generating task can be run to produce individual error and/or summary reports on some or all of the errors. Even without error logging, the device driver automatically retries recoverable errors; but after a successful recovery, the user might be unaware that the error ever occurred.

Error logging may be used on any RSX-11M/M-PLUS system that has 24K words or more of memory. It is a system generation option for RSX-11M.

Error logging reports are useful for efficient maintenance of the hardware on which the RSX-11M/M-PLUS system runs. Problems such as line noise, static discharges, or inherently error-prone media, for instance, can cause recoverable errors on systems that are otherwise functioning normally. By studying error logging reports, the user can learn to distinguish these kinds of errors from those that might be symptoms of an impending device failure. On the other hand, some recoverable errors that are insignificant in themselves might be related to other, more serious errors, the effects of which are not immediately apparent. Information contained in the reports about each error and about the status of the system when the error occurred may alert the system manager to an unforeseen hardware problem.

If the error reports seem to indicate an impending failure, the system manager should contact a DIGITAL field service engineer who will use the reports to help in diagnosing the problem. Sometimes a device fails so quickly that error log reports cannot be used to predict the failure in time to prevent it. In this case, the field service engineer can determine the cause more quickly if a report is available which describes the errors that may have occurred immediately prior to the failure.

1.2 THE FUNCTIONS OF ERROR LOGGING

If error logging is not activated on the system, the RSX-11M/M-PLUS Executive detects each hardware error, then either ignores it if it causes no immediate problem or, when appropriate, retries the function

that caused the error. The user normally has no means of knowing that such an error occurred. However, when error logging has been activated, the error logging subsystem performs the following operations:

1. Receives information from the Executive or device drivers about errors which were detected.

2. Stores the information in a file.

3. Formats the information and produces an error report.

Control of error logging is shared between routines in the Executive module ERROR and four error logging tasks. These routines and the tasks ERRLOG, PSE, SYE and ERF, work together to carry out the operations described above.

1.2.1 Error Logging Options

Error logging routines in the Executive can respond to four types of errors, depending on what options were chosen at System Generation:

1. Unexpected traps or interrupts,

2. Device errors,

3. Interrupt timeouts, and

4. Memory parity errors.

If the error logging option was generated into the Executive and the ERRLOG task was activated, all unused vectors are filled with pointers to error logging routines. Therefore, when an unusually noisy electrical environment or a static discharge causes an unexpected trap or interrupt to occur to an unused vector, or a valid interrupt to be vectored to the wrong address, Executive routines can determine and record the incorrectly used vector.

Device errors are conditions that cause a device controller to interrupt with its error bit set. When a device error occurs and error logging is active the device driver calls the Executive to record the contents of the device registers, which indicate the state of the device and its controller. In addition, these routines record information found in the actual I/O request to aid in the interpretation of the device error.

An interrupt timeout occurs when a device on which a transfer was initiated fails to interrupt within a specified amount of time. Interrupt timeouts are detected by software timers started when the transfer is initiated. The system records the same information for interrupt timeouts as it does for device errors, except no I/O activity is reported.

Some main memory uses byte parity to ensure the integrity of information. The system generates parity for both data and addresses on transfers to memory; this parity is then checked on all transfers from memory. A parity error occurs when this check fails.

1.2.2 Information Gathered

The error information gathered by the Executive routines (the state of the registers when a device error occurs, for example) provides a "snapshot" of the relevant parts of the system at the time of the error. In addition to all the system information, error routines identify the type of error and the associated device (for device errors and interrupt timeouts), record the time the error occurred and assign a sequence number to the error. The system also generates a System Service Message if the MCR SSM command was used. On RSX-11M-PLUS systems the error log report includes a CPU identification and information on whether the system has been reconfigured.

1.2.3 The Error Log File

The first two operations, error detection and the collection of error information, are continual processes done within the Executive or device drivers. When a certain amount of error information has been gathered, the Executive calls the error logging task ERRLOG to copy the information to a permanent error log file. The data may be copied to a file on the system disk or to a file on a removable disk or DECtape.

1.2.4 Formatting and Report Generation

When it is convenient or necessary to generate an error log report, the user first runs the pre-formatting task PSE to make a preliminary consolidation of the raw data contained in one or more error log files. The PSE output file becomes input to SYE, the task that actually generates the finished error logging report.

SYE is capable of generating individual error reports and/or summary reports. Among many other options, the user may specify a report that covers a certain time period, a certain device or group of devices, or a certain type of error. See Chapter 4 for a detailed description of the error reports that SYE can generate.

Because the raw error log data may be written to a removable volume, the user can generate the reports either on site or at any other RSX-11M installation that has the PSE and SYE tasks.

CHAPTER 2

HOW ERROR LOGGING WORKS

2.1 EXECUTIVE FEATURES

The RSX-11M/M-PLUS Executive is responsible for

- Detecting memory parity and unexpected trap or interrupt errors,
- Gathering volatile data that reflects, as closely as possible, the state of the system at the time of each error,
- Controlling the amount of system dynamic memory used for error logging, and activating the ERRLOG task which logs the errors

A set of common error routines performs these functions for all four types of detected errors (unexpected traps or interrupts, device errors, interrupt timeouts, and memory parity errors).

The routines use system information made available by the Executive. When an error occurs, an Executive routine notes the contents of the I/O active bitmap for subsequent analysis. The bitmap could show that the error was related to activity on some other device. If error logging of undefined traps or interrupts has been generated into the system, all unused vectors are filled with pointers to the error routines. Therefore, if a device improperly interrupts through an unused vector, an error routine is able to record the address of the vector.

The common error routines are called every time an error occurs. If the error logging option for the respective error has been generated into the system, the routines perform the following three error logging functions:

1. One routine allocates memory for error logging activity.
2. Another gathers relevant error data and formats the raw error messages.
3. The last routine queues the formatted error messages in memory in preparation for logging by ERRLOG.

Figure 2-1 is a flow chart diagram of the Executive routine activity described in this section.

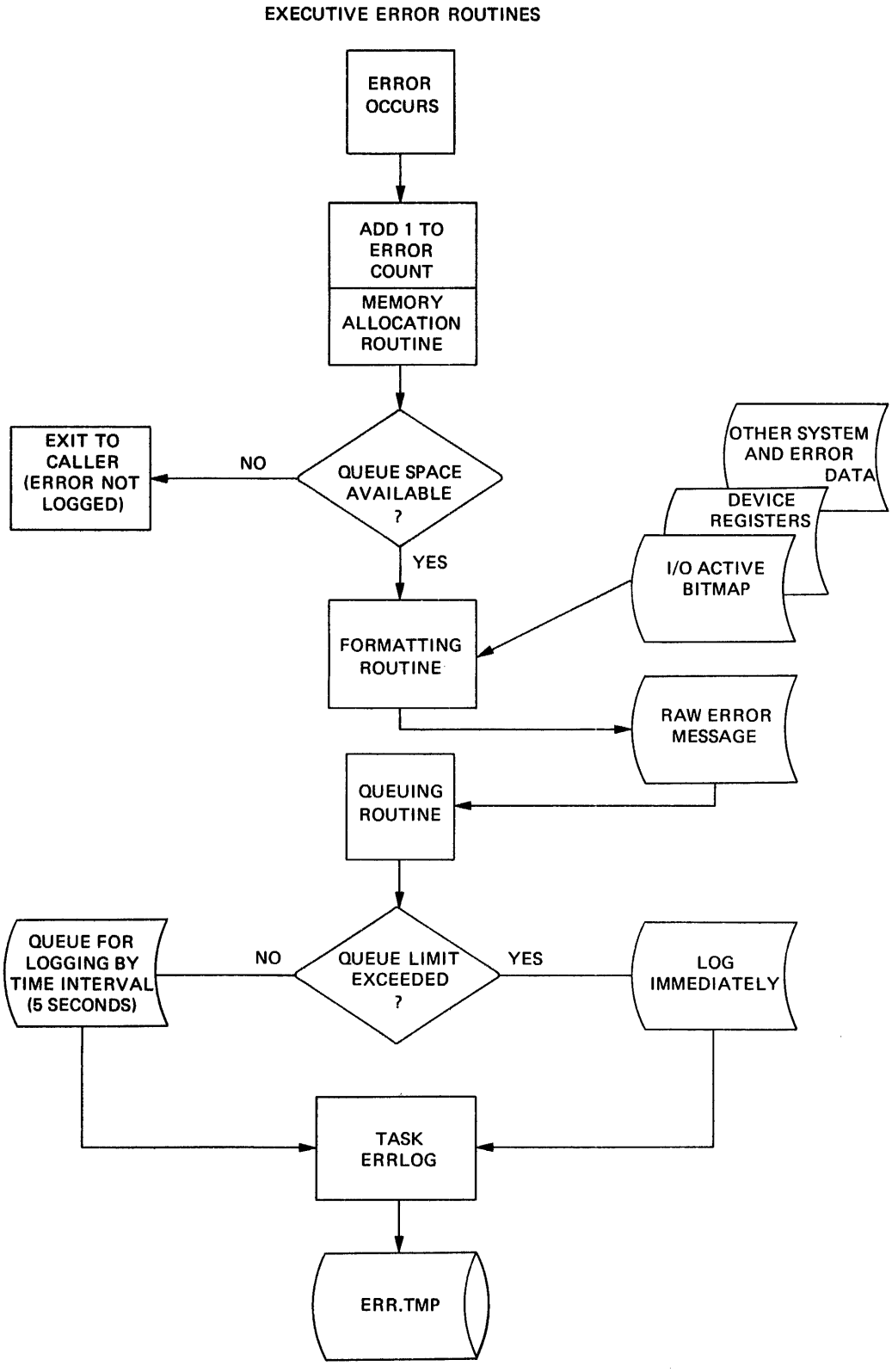

Figure 2-1 Executive Error Routines

2.1.1 Memory Allocation Routines

The memory allocation routines use the dynamic storage region (DSR) for logging error information. The routine places a limit on the amount of DSR space available. Normally each error results in a record being placed in this memory space. On device errors a counter (the error sequence number) is incremented by one. Nondevice errors, such as parity errors, do not increment the counter. The task ERRLOG is activated every five seconds to write to disk all the error records in the DSR. This frees the space for future error records. However, when a burst of errors occurs that causes the available space to be exceeded, the error records that caused the overflow are not placed in the memory space. In this case, ERRLOG is activated immediately. Subsequent errors are not logged until ERRLOG frees DSR space by writing the records at which time normal error recording resumes. The error sequence number continues to be incremented by one for each device error even if the error is not logged. When ERRLOG writes the error records to disk, the memory space is freed and normal recording of error records resumes.

2.1.2 Error Message Formatting Routines

When an error occurs, the formatting routine for that type of error creates a message containing the following information:

- The error's sequence number
- The date and time the error occurred
- A code that classifies the error
- Whether the device is mounted or dismounted
- CPU identification[1]
- Configuration information[1]
- Any additional information needed to describe the error.

Appendix B contains further details about the error message formats and formatting routines.

2.1.3 Error Message Queuing Routine

The queuing routine provides the link between Executive error logging activity and the task ERRLOG. After a raw message has been formatted, the routine enters it into a memory queue, which ERRLOG periodically writes to a file (ERR.TMP). ERRLOG scans the queue and logs the contents to a file on disk. This logging occurs in a maximum of 5 seconds after the occurrence of an error, and sooner if a burst of errors causes the queue to approach its limit. The queue containing the error data is deallocated each time ERRLOG logs its contents.

---

[1] For RSX-11M-PLUS only

2.2 TASK INTERACTION

The primary function of the error logging tasks is to maintain the error information gathered by the Executive in order to produce a report. ERRLOG preserves the Executive's information and passes it on to PSE when that task is activated. PSE prepares the data for input to SYE, which selectively generates the final, readable report. The following sections describe the file activity and error logging task interaction. Figure 2-2 demonstrates that file activity.

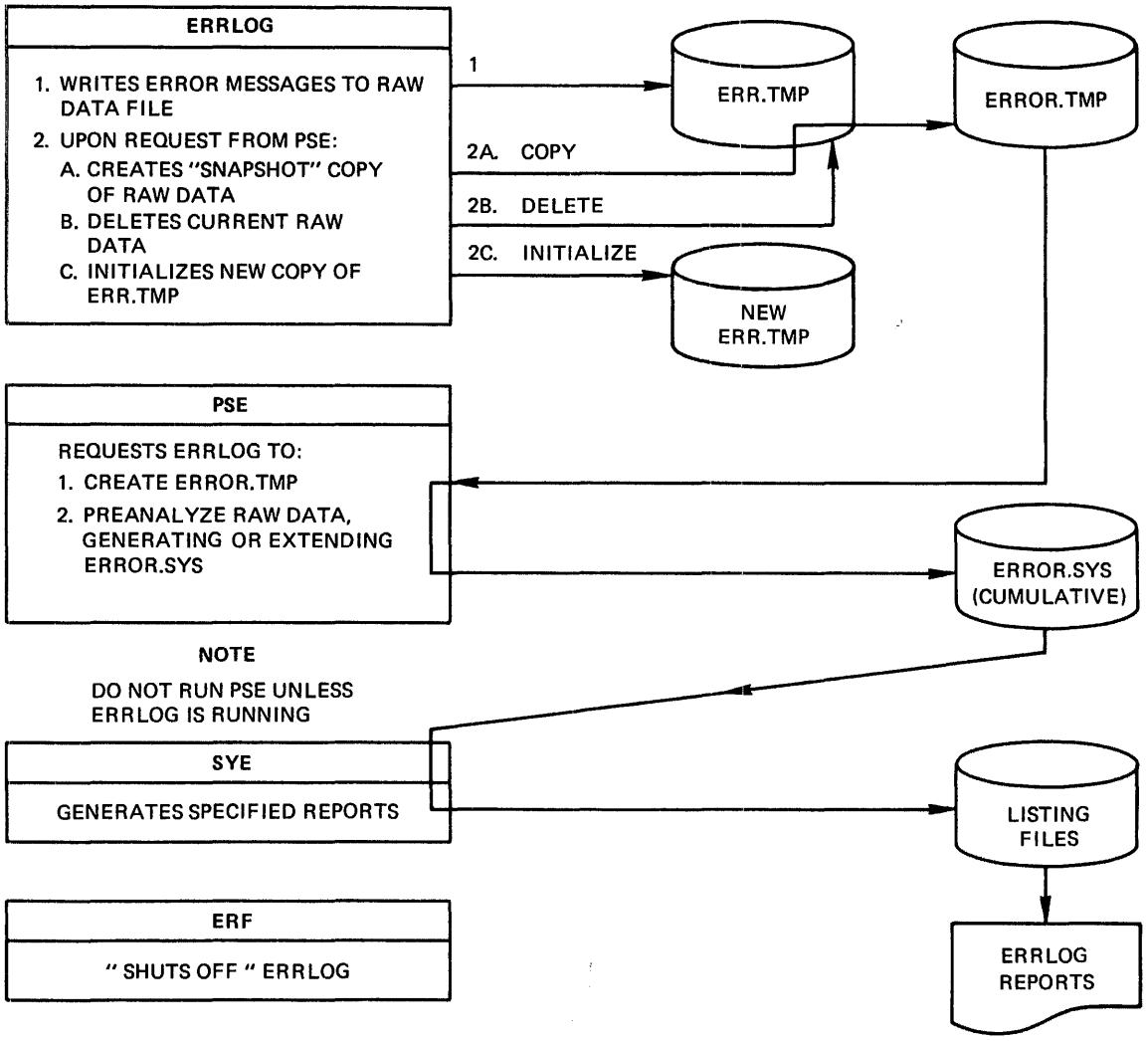

Figure 2-2 Task Interaction

The error logging subsystem on RSX-11M/M-PLUS uses various defaults for its file specifications. The default device and directory are SY0: and [1,6]. The tasks use several standard file names (including ERR.TMP, ERROR.TMP and ERROR.SYS). Users should not attempt to override these defaults (although it is possible to do so in some cases). One exception, however, is the device to which ERRLOG writes the raw data file ERR.TMP; it may be desirable to specify a dedicated, nonsystem device for this file. (This might be useful, for instance, if the ERR.TMP files have to be formatted at a different installation.)

2.2.1 ERRLOG Files (ERR.TMP and ERROR.TMP)

Each time the RUN ERRLOG command is issued, (usually at System Startup), ERRLOG creates a file called ERR.TMP. On RSX-11M and M-PLUS, ERRLOG writes system configuration records to this file for each Error Logging device. PSE uses the configuration records to determine the hardware configuration of the system for which PSE preformats the data.

ERRLOG then periodically writes to the file the raw error data it obtains from the Executive.[1] By default, the data is transferred to the file SY0:[1,6]ERR.TMP. [1,6] and ERR.TMP are the only valid values for the directory and filename, but the device field may be changed at task build time.

When the user runs PSE, ERRLOG renames the ERR.TMP data file to ERROR.TMP, makes it available to PSE and then creates a new ERR.TMP file, which includes the required system configuration information. Only one ERR.TMP file should exist at any given time.

NOTE

There may be more than one ERR.TMP file if the system has crashed or has been shut down improperly. In this case, PIP should be used to change the name to ERROR.TMP, but care should be taken to preserve the correct version number order.

Appendix B describes the format of ERROR.TMP and ERR.TMP.

2.2.2 PSE File (ERROR.SYS)

PSE (the Pre-formatter Task) requests the file ERROR.TMP from ERRLOG, preliminarily formats the raw error data, creates a file called ERROR.SYS and then deletes ERROR.TMP. If one or more ERROR.SYS files already exist, PSE appends the newly pre-formatted data to the highest version of the file. The data in ERROR.SYS is in a format that SYE can use to generate its reports.

When supplying the PSE command line, the nonprivileged user must override the output file's default device, directory and filename (SY0:[1,6]ERROR.SYS). Attempting to write to [1,6] causes a PRIVILEGE VIOLATION error when a nonprivileged user runs PSE. (Note that the input file specified in the SYE command line must be the same as the PSE output file; refer to Sections 3.2 and 3.3.)

Appendix D describes the format of the ERROR.SYS files in detail.

---

[1] On an RSX-11M-PLUS system, if you reconfigure the system, the error log file will include information on the new configuration.

2.2.3 SYE Report Files

The output file from PSE, ERROR.SYS, becomes the input file to SYE. The name of the output file that SYE produces is automatically generated. Refer to Chapter 4 for a detailed description of the contents of the output file that SYE generates. This file is the end product of the RSX-11M error logging software.

CHAPTER 3

OPERATING PROCEDURES

This chapter describes the operating procedures for the three error logging tasks (ERRLOG, PSE and SYE) and for the task that terminates error logging (ERF). Error logging must be properly generated into the system before the user can invoke these tasks (see the RSX-11M or RSX-11M-PLUS System Generation Manual). Error logging requires very little operator intervention.

After error logging has been activated either manually from a terminal or automatically by a command in the system startup command file, its operation is transparent to the user until error logging reports are generated.

Each error logging task issues a message whenever it encounters a condition (not related to errors it is logging) that halts or interferes with the task's operation. The tasks also issue informative messages (when a new file is opened in response to a request from PSE, for example). Both types of messages are listed, for each of the error logging tasks.

3.1 RUNNING ERRLOG (THE ERROR LOGGER)

ERRLOG is the only task that needs to be installed permanently for error logging to function.

To install ERRLOG, enter one of the following commands, either manually from a terminal or as an entry in the system start-up command file:

```
MCR> INS $ERL
```

The file ERL.TSK contains the ERRLOG task image. To ensure that error logging runs efficiently, ERL.TSK must be installed in a partition that meets the following criteria:

- Other tasks in the partition must be checkpointable. If ERL.TSK is installed in a user-controlled partition, all other tasks must be checkpointable.
- ERRLOG should not run in the same partition as PSE and ERF unless the partition is large and system-controlled.
- ERRLOG must not run in the same partition as F11ACP since it uses its services.

To activate error logging, enter one of the following commands, either manually from a privileged terminal or as an entry in the system start-up command file:

```
MCR>RUN ERRLOG
```

The System Startup Command File (LB:[1,2]STARTUP.CMD) may include the above command. If so, error logging is activated automatically at system startup.

Note that ERRLOG requires that the directory [1,6] exist on the error log device. The directory [1,6] must be built with full access (RWED) for PSE to run as a nonprivileged task. The system disk is the default error log device. If it does not exist, ERRLOG returns the following message:

```
ERL -- GET DIRECTORY ID FAILED, CODE -26
```

Then issue the following command to create the necessary directory:

```
MCR>UFD SY:[1,6]
```

3.2 RUNNING PSE (THE PRE-FORMATTER)

To install and run PSE, type the following:

```
>INS $PSE

>PSE
```

As distributed, PSE builds to run in an 8K partition (normally the GEN partition), providing enough space for 102. devices on a mapped system or 103. devices on an unmapped system. To increase the amount of space available to PSE for additional devices, append the /INC=xxx keyword to the Install command. (Refer to the RSX-11M/M-PLUS MCR Operations Manual for further information on the /INC switch to the MCR INStall command.) PSE needs additional space at the rate of 10(decimal) words per unit, where a unit is each device that could cause a loggable error. When running in a user-controlled (task) partition, PSE itself determines the amount of space available and then uses the amount of space it needs.

PSE is a nonprivileged task. However, you can change the ERL build file to restrict PSE from being run on a nonprivileged terminal.

The directory [1,6] must be built nonprivileged for PSE to run from a nonprivileged terminal. Then ERL must be built so that the files it creates in [1,6] (ERR.TMP and ERROR.TMP) can be read and deleted by all users (the world). When this is done, PSE, which reads and deletes ERR.TMP and ERROR.TMP can be run from any terminal. If ERL is built to generate ERR.TMP and ERROR.TMP files that cannot be read and deleted by all users, PSE must be run from a privileged terminal.

To change the way ERL creates the files, edit the ERLBLD.CMD file and change the global definition FILPRO (the file protection word).

If FILPRO=60000, all users have all privileges and the world can read and delete ERR.TMP and ERROR.TMP. In this case, PSE can run from any terminal. (See Appendix A.)

(See Chapter 4 of the IAS/RSX-11 I/O Operations Reference Manual for a description of default file protection routines and the protection mask.)

The pre-formatter task responds with the prompt PSE> and waits for the user to type a command line. The format of the PSE command line is:

```
outdev:[ufd]file.typ=indev:
```

The output file in this command is described by a standard RSX-11M/M-PLUS file specifier. The input file specifier (indev:) consists only of the input device since the filename (assigned by the task ERRLOG) is always ERROR.TMP and the UFD is always [1,6].

If you are running PSE from a nonprivileged terminal, you cannot use the default UIC [1,6] as the output UIC. This results in a privilege violation.

Both the call to PSE and the file specifiers may be included on the same line; that is:

```
>PSE outdev:[ufd]file.typ=indev:
```

The defaults for omitted fields are:

Field	Default
outdev:	SY0:
ufd	[1,6]
file	ERROR
.typ	.SYS
indev:	SY0:

To use all the defaults for both the input and output file specifiers, press the carriage return key in response to the PSE> prompt. When PSE prompts again, type CTRL/Z to return to MCR.

```
>INS $PSE               ;INSTALL PSE FROM A PRIVILEGED TERMINAL
>PSE<CR>                ;RUN PSE FROM ANY TERMINAL
PSE>^C                  ;GENERATE PRE-FORMATTED FILE
PSE>^Z                  ;EXIT FROM PSE
>REM PSE                ;REMOVE PSE
```

Note that PSE cannot be run with the RUN$ format. It must be installed from a privileged terminal and invoked as PSE <CR>.

3.3 USING SYE (THE REPORT GENERATOR)

The following sections describe the methods of invoking SYE and forms of the command line to generate the desired reports.

3.3.1 Invoking SYE

There are two methods for invoking SYE:

1. Type the following commands at a privileged terminal:

   ```
   >INS $SYE

   >SYE
   ```

2. Type the following command at any terminal:

   ```
   >RUN $SYE
   ```

The report generator task, SYE, responds with a message and the prompt SYE>, and waits for the user to type a command line.

For example:

```
>RUN $SYE
SYE>
```

3.3.2 The SYE Command Line

The format of the SYE command line is

```
outdev:[ufd]file.typ=indev:[ufd]file.typ/switchl.../switchn
```

If SYE has been installed, both the call to SYE and the file specifiers may be included on the same line; that is:

```
>SYE outdev:[ufd]file.typ=indev:[ufd]file.typ/switchl.../switchn
```

The default values for the output file specifier are:

Field	Default
outdev:	SY0:
ufd	user uic
file	ERRLOG
typ	.LST

The input file is described by a standard RSX-11M file specifier; it must be a file previously created as an output file by PSE (see Section 3.2).

The default values for the input file specifier are:

Field	Default
indev:	SY0:
ufd	[1,6]
file	ERROR
.typ	.SYS
switches	/-RP/SP

To use all the defaults for both the input and the output file specifiers, press the carriage return key in response to the SYE prompt. When SYE prompts again, type CTRL/Z to return to the system monitor. For example:

```
SYE> <CR>
SYE> ^Z
>
```

Every invocation of SYE produces summary error reports. These summary error reports are described in detail in Section 4.4 of Chapter 4. Through the use of either of two command line switches, /RP and /DV, the user can cause additional error reports to be generated along with the summary reports. An option under the SYS class of the RP switch allows you to further define the kinds of information SYE reports. As shown below, the /RP switch allows the creation of general classes of error reports while the /DV switch generates reports specific to a particular device type or device unit.

Switch	Description
/RP[:class]	The /RP switch requests individual error reports; the class parameter is used to specify a general classification of errors to be reported. If no class parameter is specified with the /RP switch, SYE reports on all classes. Class has one of the following values:
	null — All classes.
	SYS[:type] — All system errors (error logging startup, power fails, and PSE entries) which are significant to DIGITAL service groups such as field service and software support. It also reports on configuration changes on RSX-11M-PLUS. Type has one of the following values: NULL - All types of system entries PSE - Entries from PSE operation STA - Error Logging start-up MOU - device mounts and dismounts *CFG - configuration information TIM - system time changes MSG - system service message * M-PLUS only
	HDW — All hardware errors. When class is HDW, an additional parameter can be specified further defining the errors to be reported. This parameter takes the form: /RP:HDW[:type]

where

Type is one of the following:

null (no entry)	All types of hardware errors
DSK	All disk errors
MAG	All magnetic tape and DECtape errors
MEM	All cache and main memory parity errors

TMO — Interrupt time out errors from disks and tapes, in the form

/RP:TMO[:type]

Type is one of the following:

DSK	disk information only
MAG	tape information only

/SU — Create summary report

/QU — Create short summary report

/-RP or /NORP — Do not include individual error reports. SYE produces a summary report only. /-RP is the default.

/DV:dev[n] — Include in the report only those errors that occurred on a specified device type or on a specified device unit.

Besides the standard device types, you can specify:

CMM	Communications devices
UDI	Undefined interrupts
PWR	Powerfail

For example:

/DV:DK requests that error statistics on all RK03 or RK05 units be provided

/DV:DK1 requests that only the error statistics for RK03 or RK05 unit 1 be provided.

The remaining switches that can be specified on the SYE command line are:

/BEG:time-and-day

Include in the report only those errors logged after the specified time and date. The format of the time-and-date parameter is:

dd-mon-yr:hh:mm:ss

where all the numbers are decimal and leading zeros are not required. Null equals zero. The field terminator (:) must be entered for a null field. The hours (hh) field is expressed on the basis of a 24-hour clock.

Example:

/BEG:22-NOV-78:20:30:0 includes those errors that occurred after 8:30 PM on November 22, 1978.

The default is to include all errors of the specified type, no matter when they occurred.

/END:time-and-date

Include in the report only those errors logged prior to the specified time and date. The format of the time-and-date parameter is:

dd-mon-yr:hh:mm:ss

where all the numbers are decimal and leading zeros are not required. Null equals zero. The field terminator (:) must be entered for a null field. The hours (hh) field is expressed on the basis of a 24-hour clock.

Example:

/END:18-OCT-78:18:00:00 includes those errors that occurred before 6:00 PM on October 18, 1978.

The default is to include all errors of the specified type, no matter when they occurred.

/HEL[P]

The /HELP switch causes SYE to display SYE operating instructions.

To obtain this information, the user specifies the /HELP switch on its own in response to the SYE prompt as follows:

SYE>/HELP <CR>

/SP

Spool output file. This is the default. No file is printed if spooling has not been generated into the system.

SYE command line examples:

- >SYE<CR>
  SYE>

  Produces file ERRLOG.LST containing a summary report of all errors in file SY0:[1,6] ERROR.SYS. Spools ERRLOG.LST to the line printer.

- >SYE /RP<CR>
  SYE>

  Produces file ERRLOG.LST containing individual reports and a summary of all errors in file SY0:[1,6]ERROR.SYS. Spools ERRLOG.LST to the line printer.

- >SYE /DV:DB0<CR>
  SYE>

  Produces file ERRLOG.LST containing individual reports and a summary of only DB0 (RP04, RP05, RP06) errors in file SY0:[1,6] ERROR.SYS. Spool ERRLOG.LST to the line printer.

3.4 RUNNING ERF (THE SHUTDOWN TASK)

To terminate error logging, run the task ERF by typing the following pair of commands from a privileged terminal:

```
>INS $ERF

>ERF
```

Note that the command RUN $ERF is not acceptable.

CHAPTER 4

THE ERROR LOG REPORTS

4.1 GENERATING ERROR LOG REPORTS

To generate error log reports, you must first run PSE from any terminal, unless you changed the file protection mask for ERR.TMP in the ERLBLD file to restrict PSE from running from a nonprivileged terminal. In that case, the terminal must be privileged. PSE performs a preliminary formatting of the data in files created by the error logging task ERRLOG. The default PSE output file (ERROR.SYS) contains the error data which becomes the input file to SYE.

SYE, the Report Generator, can be run from any terminal. When the PSE output file is available, the user must decide what kind of reports are needed and then select the appropriate switches for the SYE command line. For example, the user can select a time frame that the report is to encompass and can specify that the report include only those errors associated with a certain device type or unit. Refer to Section 3.3 for a description of SYE operating procedures and all the available command line switch options.

SYE produces an error report in the form of a printed listing or a listing file or both. By default, SYE creates a file called ERRLOG.LST on the system disk; this file contains summary information on all of the errors recorded in the ERROR.SYS file. Both PSE and SYE are generated if you select these options in Phase 3 of System Generation.

4.2 INDIVIDUAL ERROR REPORTS

Error logging detects and records four kinds of errors: unexpected traps or interrupts, device errors, interrupt timeouts, and memory parity errors (refer to Section 1.2.1). This section illustrates sample reports on each kind of error.

4.2.1 Reports For Unexpected Traps or Interrupts

Figure 4-1 is a report of an interrupt through an unused vector. Such interrupts are also called traps in SYE reports. The description of the report refers to each line of text by number, where the first line described is line 1.

```
                                COMMAND

 1  ************************* ENTRY 12. ****************************
 2  UNEXPECTED TRAP OR INTERRUPT
 3  LOGGED  28-JAN-79 16:24:00   ON -- CPA--  Configuration Entry #12[1]
    ************************************************************************

 4  SOFTWARE STATUS
 5          VECTOR WHERE TRAP OR INTERRUPT OCCURRED  200
 6          PC      PRIOR  THIS ERROR               177000
 7          PSW     PRIOR  THIS ERROR               000004
 8          ERRORS  MISSED WHILE THIS ERROR LOGGED  0

 9  DEVICE I/O ACTIVITY
10          204
```

[1] RSX-11M-PLUS only

Figure 4-1 Report of an Unexpected Trap or Interrupt

- Line 1 gives the record's entry number. The first error to occur after error logging has been initialized is assigned an entry number of 1. The entry number is then incremented by 1 every time a loggable error occurs.
- Line 2 describes the type of error.
- Line 3 gives the date and time the error occurred, and, on RSX-11M-PLUS systems, identification information on the CPU and system configuration changes.

The next block of text (lines 4 through 8) contains information about the software status of the trap or interrupt:

- Line 5 provides the address of the vector through which the interrupt occurred.
- Line 6 is the value of the program counter immediately prior to the error.
- Line 7 is the processor status word immediately prior to the unexpected trap or interrupt.
- Line 8 notes the number of errors that occurred while the system was logging this error. (This count permits the detection of bursts of illegal interrupts possibly caused by UNIBUS noise corruption.)

The last block of data, headed "DEVICE I/O ACTIVITY", lists all the vectors with I/O activity on the UNIBUS at the time of the error. The list includes the interrupt address of the device that caused the error.

4.2.2 Device Error Reports

Figure 4-2 is a typical error log report on a device error. (The figure also serves as a sample interrupt timeout report, since the report format is identical for both types of error.) The following description of the device error report refers to each line of text by number, where the first line described is line 1.

```
 1  ************************** ENTRY 36. ****************************
 2  DEVICE HARDWARE ERROR
 3  LOGGED   17-FEB-79 13:39:59   ON CPA -- Configuration Entry #2[1]
    ******************************************************************

 4  UNIT IDENTIFICATION
 5          UNIT LOGICAL NAME       DK0
 6          UNIT PHYSICAL NAME      DK0  (CONTROLLER-0    UNIT-0)
 7          DEVICE TYPE             RK05/RK05F/RK03 (DISK)
 8          VOLUME LABEL

 9  SOFTWARE STATUS
10          TASK NAME               ...VFY
11          TASK UIC                1,1
12          TASK START ADDRESS      373400
15          IO FUNCTION ISSUED      READ

16  DEVICE REGISTERS
17          RKDS    004723
18          RKER    000002  CHECKSUM ERROR
19          RKCS    100744  READ
20          RKWC    000000
21          RKBA    132211
22          RKDA    000023

23 MEDIA ADDRESS            AT IO START     AFTER ERROR
24          CYLINDER                0       0
25          TRACK                   1       1
26          SECTOR                  2       2
27          LOGICAL BLOCK           17      17

28  ERROR DIAGNOSIS
29          NOT RECOVERED AFTER 9. RETRIES

30  DEVICE I/O ACTIVITY
31          220
```

[1] RSX-11M-PLUS only

Figure 4-2 Report of a Device Hardware Error

The first two lines of text, offset by two star lines, give header information about the error:

- Line 1 gives the record's entry number and identification information on the CPU and configuration for RSX-11M-PLUS systems.
- Line 2 describes the type of error (DEVICE HARDWARE ERROR).
- Line 3 gives the date and time the error occurred.

The next block of data (UNIT IDENTIFICATION) describes the erring device:

- Line 5 (UNIT LOGICAL NAME) shows the device mnemonic (DK) and unit logical number (0). The unit logical number ranges from 0 to n, where n is an octal number representing the total number, minus one, of such devices supported by all controllers at a particular installation.

- Line 6 (UNIT PHYSICAL NAME) shows the device mnemonic, the physical unit number, and the controller number. The physical unit number can range from 0 to 7 for each controller.

- Line 7 (DEVICE TYPE) shows the DIGITAL name for the device (RK05/RK05F/RK03).

- Line 8 provides the volume label identification for RSX-11M systems. (If there is no volume label, the field remains blank.)

The next block of data (SOFTWARE STATUS) describes the user task that initiated the I/O request that failed.

- Line 10 provides the 6-character name of the task that initiated the erring I/O request (...VFY).

- Line 11 is the User Identification Code (UIC) under which the task was running.

- Line 12 shows the task's physical start address, that is, the base address of the task in memory.

- Line 15 describes the I/O function requested by the task.

The report then describes, in lines 16 through 22, the state of the device registers at the time of the error. The first column lists the register names; the second shows the contents of the registers. The device register names are those used in the PDP-11 Peripherals Handbook. The third column, following some registers (lines 18 and 19) explains the meaning of the error bits in register RKER and RKCS, respectively.

NOTE

The function that failed, as described by the register contents in lines 16-22 may differ from the function requested by the task.

For example, if a task requests a WRITE to a DECtape, the driver must perform a READ BLOCK NUMBER function, which could then cause an error.

The next block of text (MEDIA ADDRESS) shows the physical disk address at the start of the I/O operation and at the time of the error (this section does not appear for DECtape, magnetic tape, and cassette errors). Line 27 (LOGICAL BLOCK) specifies the logical block number requested at I/O start time and the logical block number computed from the device registers at error time.

The next block, (ERROR DIAGNOSIS), (lines 28 and 29) show that the error was not successfully recovered and indicates how many times the device driver retried the operation. This line alternatively shows the number of times the driver retried the error if recovery was successful.

The last block of text (lines 30 and 31) lists the vectors with active data transfers on the UNIBUS at the time of the error. The list includes the interrupt vector address of the device that caused the error. The number on the second line (220) is the vector address

through which a transfer complete or data error interrupt would be fielded for the device.

The block headed "VECTORS WITH ACTIVE IO" does not appear if there was no I/O activity when the error occurred. Also, error reports on interrupt timeouts do not include this information because the long time element involved makes it irrelevant.

4.2.3 Device Interrupt Timeout Reports

Interrupt timeout reports have the same format as device error reports. Refer to Section 4.2.2 for a description of the various entries.

4.2.4 Memory Parity Error Reports

Figure 4-3 illustrates a report of a memory parity error. The following description of the report refers to each line of text by number, where the first line described is line 1.

```
 1  ****************************** ENTRY 524. ***********************
 2  MEMORY PARITY ERROR
 3  LOGGED   5-JUN-79 13:03:36
    ************************************************************************

 4  SOFTWARE STATUS
 5          TASK NAME                TT61
 6          TASK START ADDRESS       721200
 7          TASK SIZE (WORDS)        7000
 8          PC  PRIOR TO THIS ERROR  121212
 9          PSW PRIOR TO THIS ERROR  170000
10          PARTITION NAME           GEN
11          PARTITION START ADDRESS  721200

12  MEMORY REGISTERS  (11/70 TYPE)
13          LOW ERROR ADDRESS        122410
14          HIGH ERROR ADDRESS       000003
15          MEMORY SYSTEM ERROR      104404   MAIN MEMORY EVEN WORD
16                                            ERROR IN MAINTENANCE
17                                            CPU ERROR
18                                            CPU ABORT
19          MEMORY CONTROL           000001   DISABLE TRAPS
20          MEMORY MAINTENANCE       000000
21          MEMORY HIT/MISS          000043

22  ERROR DIAGNOSIS
23  MEMORY PARITY ERROR IN LOCATION 722410

24  WITH MEMORY MANAGEMENT REGISTER 5 ACTIVE IN USER MODE
```

Figure 4-3 Report of a Memory Parity Error

The first three lines of text, offset by two star lines, give header information about the error:

- Line 1 gives the record's entry number.

- Line 2 describes the type of error (MEMORY PARITY ERROR).
- Line 3 gives the date and time the error occurred, and supplies identification on the CPU and system configuration for RSX-11M-PLUS systems.

The next block of data (SOFTWARE STATUS) describes the user task that was active when the error occurred:

- Line 5 shows the name of the task.
- Line 6 gives the task's physical start address; that is, the base address of the task in memory.
- Line 7 gives the size of the task in words.
- Lines 8 and 9 give the task's Program Counter (PC) and Processor Status word (PSW) before the error occurred.
- Line 10 and 11 provide the name of the partition containing the task, and the physical start address of the partition.

The next block (MEMORY REGISTERS) describes the state of the memory registers at the time of the error. The register names listed in the first column are those used in the PDP-11 Processor Handbooks. The second column shows the contents of the registers; and the third column describes individual bits in the MEMORY SYSTEM ERROR register and the MEMORY CONTROL register.

PDP-11/40 and /45 Processors Only - When a report describes a memory parity error that occurred on a PDP-11/40 or /45 processor, the report displays the address as well as the contents of each register.

Lines 22 through 24 provide an error diagnosis:

- Line 23 gives the physical location in memory in which the parity error occurred. PDP-11/40 and /45 Processors Only - When the parity error occurred on a PDP-11/40 or /45 processor, this line gives the range of addresses in which the error occurred, rather than a precise location. For example, the line might read:

  PARITY ERROR IN LOCATIONS 210000 TO 213777

Line 24 appears for mapped systems only:

- This line gives the number of the memory management register group active at the time of the error and the operating mode of the task that was active when the error occurred.

4.3 OPTIONS FOR SYE RP SWITCH

Figures 4-4 through 4-8 are examples of the reports SYE generates with various options for the RP switch.

```
1 ***************************** ENTRY 197. ******************************
2 DEVICE MOUNTED MESSAGE
3 LOGGED   22-DEC-78 01:35:52 ON CPA -- Configuration Entry #2[1]
  ****************************************************************

4 DEVICE MOUNTED-- DK0:DBLWAK  UIC=[1,1]   UCB ADDR.=022252
```

[1] RSX-11M-PLUS only

Figure 4-4 Device Mounted Report

Line 2 shows the option with the RP:SYS switch which caused SYE to generate the message. In this case, the SYE command could have included:

```
/RP:SYS:MOU
/RP:SYS
   or
/RP
```

Line 3 gives the day and time the action occurred and, on RSX-11M-PLUS, a message on the system configuration.

Line 4 gives specific information on which device was mounted.

```
1 ***************************** ENTRY 204. ******************************
2 SYSTEM CONFIGURATION MODIFIED
3 LOGGED   22-DEC-78 03:20:11 -- CONFIGURATION ENTRY #2.[1]
  ****************************************************************

4 TT56:   YHC15:  ONLINE,ACCPATH,MTD-LGD,DRIVER
```

[1] RSX-11M-PLUS only

Figure 4-5 Configuration Change Report

Line 2 shows the option with the RP:SYS switch that caused SYE to generate this message. The following SYE commands can generate this report:

```
/RP:SYS:CFG
/RP:SYS
/RP
```

Note that this switch (CFG) only works on RSX-11M-PLUS systems, which include re-configuration.

Line 3 shows the day and time the action occurred and the entry number for the last configuration change.

Line 4 shows the actual change that was made in the system.

```
1 ***************************** ENTRY 196. ******************************
2 SYSTEM TIME CHANGE
3 LOGGED  22-DEC-78 01:00:33
  ***********************************************************************
```

Figure 4-6 Time Change Report

Line 2 shows the option with the RP:SYS switch that caused SYE to generate this report. The following SYE commands can generate this report:

```
/RP:SYS:TIM
/RP:SYS
/RP
```

```
1 ************************** ENTRY 3. ******************************
2 SYSTEM SERVICE MESSAGE
3 LOGGED   11-JUN-79 10:16:17
  ***********************************************************************

4 TASK INITIATING MESSAGE:  ...MCR
5 THE MESSAGE IS BELOW:



6 THIS TEXT WAS INSERTED IN THE ERROR LOG REPORT BY AN MCR SSM COMMAND
```

Figure 4-7 System Service Report

Line 2 shows the option with the RP switch that caused SYE to a message.

Line 3 shows the day and time the action occurred.

Line 4 shows the task that initiated the message; in this case MCR

Line 6 is the message text.

The following SYE command can generate this report.

```
/RP:SYS:MSG
/RP:SYS
/RP
```

4.4 SUMMARY REPORTS

Figure 4-8 is an example of the summary section of a report. This part of the report contains summary information about all the errors described in the report as a whole (but not necessarily about all the errors in the input file to SYE).

```
 **********************************************************************
                  SYSTEM ERROR REPORT SUMMARY TOTALS
 **********************************************************************

  2  DEVICE ERRORS                     80.
  3  MEMORY PARITY ERRORS              0.
  4  CACHE PARITY ERRORS               0.
  5  INTERRUPT TIMEOUT ERRORS          0.
  6  UNEXPECTED TRAPS OR INTERRUPTS    0.
  7  POWER FAILS                       0.
  8  ENTRIES MISSING (SATURATION)      0.
  9  TOTAL ERROR CP0 COUNT [1]         80.


 10          COMMAND LINE USED         ERK05.LST/-SP=[304,324]DKDBDS.SYS
 11          INPUT FILE                SY0:[304,324]DKDBDS.SYS;1
 12          OUTPUT FILE               SY0:[304,324]ERK05.LST;2
 13          DATE OF FIRST ENTRY       17-FEB-77 05:58:16
 14          DATE OF LAST ENTRY        17-FEB-77 15:39:08
 15          ENTRIES PROCESSED         94.
 16          UNRECOGNIZABLE ENTRIES    0.




 **********************************************************************
 17              DETAILED DEVICE ERROR REPORT SUMMARIES
 **********************************************************************




 18  PROCESSOR[1]              CPA
 19  UNIT LOGICAL NAME[1]      DS2
 20  UNIT PHYSICAL NAME[1]     DS2       CONTROLLER-0  UNIT-2
 21  DEVICE TYPE[1]            RS04
 22  VOLUME LABEL[1]           JEPBACKUP
 23                                                   [HARD]   [SOFT]
 24                            DATA LATE              0.       35.



 25  UNIT LOGICAL NAME         DK0
 26  UNIT PHYSICAL NAME        DK0       CONTROLLER-0 UNIT-0
 27  DEVICE TYPE               RK05/RK05F/RK03
 28                                                   [HARD]   [SOFT]
 29                            MOTION/SEEK            0.       1.
 30                            CHECKSUM OR CRC        9.       0.
 31                            DATA LATE              9.       0.

[1] RSX-11M-PLUS only
```

Figure 4-8 Summary Error Report

Lines 2 through 9 of the summary report provide various error statistics:

- Line 2 is the number of device errors described in this report.
- Line 3 records the total number of memory parity errors described in the report.
- Line 4 shows the total number of cache parity errors.
- Line 5 is the number of interrupt timeouts described in this report.
- Line 6 is the number of unexpected traps or interrupts.
- Line 7 gives the number of power fails.
- Line 8 is the number of errors that were not logged because another error was already being recorded. Lines 10 through 16 of the report summarize details of the execution of SYE.
- Line 9 provides the total errors which occurred on each processor for an RSX-11M-PLUS system.
- Line 10 is the command line issued to SYE.
- Line 11 describes the complete file specifier of the input file and line 12 describes the complete file specifier of the output file.
- Lines 13 and 14 give the dates and times of the first and last entries in the input file.
- The number of entries formatted in the report appears in line 15.
- Line 16 is the number of unrecognizable errors encountered by SYE. An unrecognizable error is any entry that the current version of SYE cannot format. This situation can occur for one of the following reasons:

  1. An old version of SYE has been run.
  2. The entry refers to a device not supported by SYE. Such an entry may be encountered if the site has implemented error logging on a device SYE does not recognize.
  3. SYE has encountered a data structure (field format) error. Such encounters indicate that the wrong version of the pre-formatter PSE was used.

Line 17 marks the beginning of a series of detailed device error report summaries. Each such summary provides error statistics relating to an individual device. Figure 4-8 shows two such summaries, the first on lines 18 through 24, and the second on lines 25 through 31.

- Line 18 shows the processor (on RSX-11M-PLUS systems) for which the report was generated.
- Lines 19 and 25 show the unit logical name.

- Lines 20 and 26 give the physical unit numbers and controller numbers of the respective devices.
- Lines 21 and 27 show the DIGITAL name for the device.
- Line 22, on RSX-11M-PLUS systems, provides the Volume Label for the device.

The remaining sections in each detailed device error summary (lines 21 and 24 in the first instance and lines 28 through 31 in the second) show the number of hard (non-recovered) and soft (recovered) errors described in this report. The types of errors which can appear in these sections, examples of which appear on lines 24 and 29 through 31, are described below.

Heading	Description
1 CHECKSUM OR CRC	The controller detected either a data checksum error or a hard error-check correction error.
2 CONTROLLER PARITY	The hardware controller detected a parity error within itself.
3 DATA LATE	The controller detected a data error caused by one of two possible situations: • Data was ready for memory when memory had not emptied its hardware buffer (silo), or • The controller was ready for data, but memory's silo had no data to be written.
4 DATA PARITY	The controller detected a data parity error.
5 DEV-TO-MEM PARITY	The controller detected a parity error while the controller was performing a Write or Write-Check command.
6 DRIVE	A drive specific error, for example a drive timing error, occurred.
7 DRIVE SELECT	A drive selection error, such as the selection of multiple heads, has occurred.
8 HEADER	The controller detected a unit error that prevents a successful data transfer. Specific errors include search errors, header compare errors, and format errors.
9 INTERRUPT TIMEOUT	An initiated operation did not cause an interrupt within a specified time interval.
10 MOTION/SEEK	The controller detected mechanical problems on a device unit, which cause the unit to produce errors such as seek incomplete, seek error, or operation incomplete.

Heading	Description
11 NONEXISTENT MEMORY	The controller detected a bus timeout error.
12 POWER ERROR	The controller detected an error related to power difficulties.
13 SOFTWARE ERROR	Corrupted service routines caused a controller error.
14 UNIT UNSAFE	The controller detected an error that caused a selected device to be unusable.
15 WRITE CHECK	A write check error occurred.
16 WRITE LOCKED	A write operation was attempted on a write-locked unit.
17 UNCATEGORIZED ERRORS	Any error that does not fit in one of the above categories.

CHAPTER 5

ERROR LOGGING TASK MESSAGES

5.1 ERRLOG MESSAGES

5.1.1 Informational Messages

The following two messages are informational messages issued during ERRLOG's normal operation.

ERL -- ERROR LOG INITIALIZED

The file ERR.TMP has been opened and initialized successfully. The message occurs either when the RUN ERRLOG command is issued. or after ERRLOG has renamed the current ERR.TMP file to ERROR.TMP and opened a new ERR.TMP.

ERL -- LOGGING ENDED AFTER ddd ERRORS

ERRLOG issues this message when error logging is shut down, where "ddd" is the number of errors that were logged to the current file. This count will have been zeroed if PSE had received a file from ERRLOG while logging was active; therefore it does not necessarily indicate the total number of errors logged since ERRLOG was initialized. Note that if ERRLOG is shut down and an error occurs, the error count is updated but the error is not logged.

5.1.2 Error Messages

The error logging subsystem terminates in response to an error condition unless the description of the associated error message states otherwise. All error codes are decimal numbers. See the IAS/RSX-11 I/O Operations Reference Manual, Appendix I, for a description of the error codes.

Table 5-1 illustrates sources for error logging messages.

Table 5-1
Source of Error Codes in ERRLOG Messages

Source	Kind of number
File Control Service (FCS)	Negative decimal number
Driver/File Control Primitives (FCP)	Positive decimal number
File Descriptor Block (FDB)	Positive decimal number

ERL -- ASSIGN LUN FAILURE

Explanation: An error occurred when ERRLOG tried to assign LUN 4 while creating file ERR.TMP. Logging is not initialized.

User Action: Retry the ERRLOG operation.

ERL -- DEVICE FULL. MOUNT NEW VOLUME OR
"REA ERRLOG 4 DDU:",THEN "RUN ERRLOG"

Explanation: While trying to write an error to the file, ERRLOG found the device full. The buffer containing the most recent error logged is requeued in memory, and error logging is shut down.

User Action: In order to reinitialize logging, either a different device with available space must be used, or the full volume must be dismounted, a new one mounted and the task ERRLOG restarted; the system will then reinitialize error logging. (There may be errors queued in the dynamic pool ready to be logged.)

ERL -- DEVICE RECORD PUT ERROR, CODE ddd

Explanation: During ERR.TMP initialization, a file system error was detected in writing a device configuration record.

ERL -- ERROR ON REOPEN, CODE ddd

Explanation: ERRLOG was unable to open an existing, initialized ERR.TMP file in order to write an error message to it.

ERL -- ERR.TMP NOT RENAMED (ERR.TMP: ddd, ERROR.TMP: ddd)

Explanation: ERRLOG failed to rename the ERR.TMP file either when passing the file contents to PSE or when error logging was being shutdown. The codes are from the respective files' File Descriptor Blocks (FDBs) (the code may be 0 for ERROR.TMP, indicating the problem was only with ERR.TMP). In this case, error logging is not shutdown since a new file will be created.

User Action: Once the reason for the rename failure is determined, the file may be renamed to ERROR.TMP using PIP, and subsequently formatted (see the RSX-11M Utilities Manual).

ERL -- ERR.TMP OPEN FAILURE, CODE ddd

Explanation: The file system reported a failure during initialization when ERRLOG tried to create the file ERR.TMP.

ERL -- FILE CLOSE ERROR, CODE ddd

Explanation: After ERRLOG wrote initialization information or error records, FCS returned an error when it tried to close ERR.TMP. The file will be locked and truncated; error records may therefore be missing from the end of the file.

User Action: Use PIP to unlock the file for formatting or for further logging if the error is not persistent (see the RSX-11 Utilities Manual).

ERL -- GET DIRECTORY ID FAILED, CODE ddd

Explanation: Failure during creation of a new ERR.TMP file. If the code is -26, the directory [1,6] does not exist on the system disk. (Refer to Section 3.1.)

User Action: Create UFD and retry operation.

ERL -- INITIALIZATION PUT ERROR, CODE ddd

Explanation: During file initialization, an error occurred on the first attempt to write to the ERR.TMP file.

ERL -- MARKED FOR REMOVE, LOGGING NOT STARTED

Explanation: This message will appear if the command RUN $ERL has been issued to start up error logging. ERRLOG must be permanently installed; therefore it may not be executed by the default "install/request/remove" method of MCR.

User Action: Restart error logging using commands in Section 3.1 of this manual.

ERL -- NO ERROR FILE

Explanation: This error is unlikely to occur. It indicates that some task has zeroed an internal file ID pointer. ERRLOG exits when it encounters this situation.

User Action: Rerun ERRLOG to activate error logging.

ERL -- OUTPUT ERROR, CODE ddd

Explanation: This error was caused by an attempt to write data to the file ERR.TMP after the occurrence of a loggable error.

ERL -- TASK NAME NOT "ERRLOG"

Explanation: This message appears when the user tries to initialize ERRLOG after the task has been installed incorrectly.

User Action: Install the error logging task under the name "ERRLOG".

ERL -- TERMINAL NOT PRIVILEGED

Explanation: The terminal from which ERRLOG is initialized must be privileged.

User Action: Log on a privileged terminal and retry the operation.

5.2 PSE MESSAGES

PSE -- COMMAND STRING PARSE ERROR

Explanation: PSE encountered a syntax or semantic error when examining an input command line.

User Action: PSE prompts for further input. Retype the corrected command line.

PSE -- DELETE ERROR ON INPUT FILE

Explanation: When PSE finished processing the input file ERROR.TMP, the task was unable to delete it.

User Action: Use PIP to delete ERROR.TMP.

PSE -- ERROR LOGGER DID NOT PROVIDE A FILE

Explanation: PSE was able to communicate with ERRLOG, but ERRLOG did not make a file available within 30 seconds. This message is also displayed when the RUN$PSE command format is used to indicate that PSE cannot run.

User Action: Run PSE again in the form: INS $PSE <CR> PSE<CR>

PSE -- 1ST RECORD WAS NOT AN INITIALIZATION RECORD FILE CANNOT BE PRE-ANALYZED

Explanation: The first record of a file must be an initialization record. Both the input and the output files are closed. The input file is not deleted.

User Action: Rerun PSE, making sure you are giving PSE the correct input file. In other words, the file must be output from ERRLOG and not output from PSE or SYE. If the error occurs again, delete the oldest version of ERROR.TMP.

PSE -- INPUT FILE ERROR

Explanation: An error was encountered when PSE tried to open or obtain data from the input file ERROR.TMP. The input and output files are closed. The input file ERROR.TMP is not deleted.

User Action: Rerun PSE. If the error occurs again, delete ERROR.TMP.

PSE -- OUTPUT DEVICE IS FULL

Explanation: The output device became full while PSE was writing to the output file. Both the input and output files were closed. ERROR.TMP is not deleted.

User Action: Select another volume for the output file.

PSE -- OUTPUT FILE ERROR

Explanation: An error was encountered while PSE was accessing the output file. Both the input and output files are closed. ERROR.TMP is not deleted.

User Action: Run PSE with output file sent to an unprivileged UFD.

PSE -- RECORD SIZE WAS INCORRECT

Explanation: When PSE reads a record from the input file, it verifies the size of the record. If the size is wrong, it ignores the record and continues to process the rest of the input file.

PSE -- TOO MANY DEVICE DESCRIPTOR ENTRIES. RECOMMEND THAT PSE RUN IN A LARGER PARTITION
DEVICE = device
IDENTIFIER = id

Explanation: PSE has run out of storage space in which to hold descriptor entries for the device named in the message. PSE continues to process the input file, but it is unable to process any more errors for the named device.

User Action: In the future, run PSE for this configuration in a larger partition, or specify the /INC=xxx switch when PSE is installed in a system controlled partition. PSE requires 10 (decimal) words of space for each device for which errors are to be logged.

PSE -- UNABLE TO CLOSE THE INPUT FILE

Explanation: PSE is unable to close the input file ERROR.TMP. The file is not deleted.

PSE -- UNABLE TO CLOSE THE OUTPUT FILE

Explanation: PSE is unable to close the output file. The message generally indicates that the output device is full and that the last block of the file will be missing.

User Action: Make more space available on your output device.

PSE -- NO ERROR.TMP FILE PROCESSED

Explanation: PSE cannot locate any files named ERROR.TMP.

User Action: Verify that the correct input device was specified or that ERRLOG has been initialized.

PSE -- UNKNOWN DEVICE OR CONTROLLER IDENTIFIER
IDENTIFIER = id

Explanation: PSE has received a device descriptor error entry that does not match any device described in the device tables. PSE ignores that entry record and continues to process the rest of the file.

PSE -- UNABLE TO RECOGNIZE ERROR ENTRY CODE
ENTRY CODE = code

Explanation: PSE must recognize the entry code contained within each record in order to know how to process the record. If PSE does not recognize the entry code, it ignores the record and continues to process the rest of the input file.

PSE -- UNABLE TO REQUEST A FILE FROM ERRLOG

Explanation: PSE tried to request ERRLOG to make a file available for formatting. This error indicates that ERRLOG has not been initialized.

User Action: Run ERRLOG and then run PSE again.

5.3 SYE MESSAGES

5.3.1 Command Line Errors

The messages described in this section occur when SYE encounters an error in the command line.

SYE -- COMMAND STRING ERROR
ERROR NUMBER x

Explanation: SYE has encountered an error (other than end of file) when trying to obtain a command line.

User Action: Refer to the IAS/RSX-11 I/O Operations Reference Manual, Appendix I, for an explanation of the error code "x", a decimal number.

SYE -- COMMAND STRING ERROR
text

Explanation: SYE encountered an error when it tried to parse the command line. "text" is the command line that contains the error.

User Action: Check syntax and re-enter command.

SYE -- COMMAND STRING SYNTAX ERROR
text

Explanation: SYE encountered an error in the syntax of the command line; "text" is the erring portion of the command line.

User Action: Reenter the command, using the correct syntax.

SYE -- ILLEGAL REPORT SWITCH
/REPORT:text

Explanation: Illegal values ("text") have been supplied with the report (/RP) switch.

User Action: See Section 3.3.

SYE -- ILLEGAL SWITCH COMBINATION
NO OUTPUT CAN BE GENERATED

Explanation: The SYE command line specified an illegal switch combination.

User Action: Reenter the command line.

SYE -- NOT CATEGORIZED

Explanation: There was a hard or soft error on the specified device, but the error did not fit in any of the 16 defined categories.

SYE -- NO ARGUMENT FOR DEVICE SWITCH

Explanation: The SYE command line did not specify a group of devices or a specific device in the device switch (that is, /DV).

User Action: Reenter the command line.

5.3.2 File Service Errors

The following messages are caused by file failures. Each message contains the line "FATAL ERROR - x" where x is a decimal number that represents an error code returned by File Control Services (FCS). See the IAS/RSX-11 I/O Operations Reference Manual, Appendix I.

SYE -- DEVICE ERROR INPUT FILE
FATAL ERROR - x.
filename

Explanation: An error (other than end of file) occurred when SYE attempted to read data from the input file. SYE closes all files and prompts for the next command line.

SYE -- DEVICE ERROR INPUT FILE
NO SUCH FILE
filename

Explanation: The input file specified in the SYE command line does not exist. SYE closes all files and prompts for the next command line.

SYE -- DEVICE ERROR OUTPUT FILE
FATAL ERROR - x.
filename

Explanation: SYE encountered an error when attempting to write data to the output file. SYE closes all files and prompts for the next command line.

SYE -- OPEN FAILURE ON INPUT DEVICE
FATAL ERROR - x.
filename

Explanation: SYE encountered an error when it tried to open the input file. SYE closes all files and prompts for the next command line.

SYE -- OPEN FAILURE ON OUTPUT DEVICE
FATAL ERROR - x.
filename

Explanation: SYE encountered an error when it tried to open the output file. SYE closes all files and prompts for the next command line.

SYE -- SUMMARY TEMP FILE I/O ERROR REPORT CONTINUES WITHOUT SUMMARIES

Explanation: SYE encountered an error while reading or writing to file SYESUM.TMP. This file is a temporary file created when SYE runs out of available work space in memory.

5.3.3 Additional Messages

SYE -- CONFIGURATION BUFFER FILLED

Explanation: Due to a large number of configuration records in the input file, a buffer in SYE is filled to capacity and the configuration information reported may be incomplete.

SYE -- TEMPORARY FILE ERROR
REPORT CONTINUES WITHOUT SUMMARIES

Explanation: SYE was unable to produce a complete summary report because it encountered an error in a temporary file containing data for the device summary reports. SYE produces a report without the device summaries.

User Action: None required

SYE -- x. PAGES IN REPORT

Explanation: SYE displays this message when it has completed its formatting of the complete report. "x." is the decimal number of pages in the report.

User Action: None required.

5.4 ERF MESSAGES

The shutdown task ERF always displays a message in response to a request to run; the message is either an error message or a confirmation of shutdown.

ERF -- "ERRLOG" NOT INSTALLED

Explanation: The task ERRLOG has not been installed.

User Action: Install ERL and run again.

ERF -- ERROR LOGGING NOT STARTED

Explanation: Error logging was not initialized, and therefore does not need to be shutdown.

ERF -- REQUESTED "ERRLOG" TO STOP LOGGING

Explanation: The normal confirmation message. ERF has exited and ERRLOG should display its own termination message shortly.

User Action: None. This is the ERF success message.

ERF -- TERMINAL NOT PRIVILEGED

Explanation: ERF must be requested from a privileged terminal.

User Action: Run ERF from privileged terminal.

APPENDIX A

MODIFYING ERL AND SYE AT TASK BUILD

A.1 ERL

The command file ERLBLD.CMD contains instructions to the task builder to create the ERRLOG task. There is a global definition which may be changed which affects the way the PSE runs.

This definition (FILPRO) is the file protection mask for ERL-generated files [1,6]ERR.TMP and [1,6]ERROR.TMP. These are files that are later read and deleted by PSE.

Thus if the file protection mask extends read and delete privileges to the world, PSE can be run from a nonprivileged terminal.

FILPRO=60000 extends all privileges to system, owner and group and read and delete privileges to the world.

Change the value of FILPRO to 170000 to require PSE to run from a privileged terminal. See the IAS/RSX-11 I/O Operations Reference Manual, Chapter 4, the section describing default file protection routines for a complete description of the file protection mask.

A.2 SYE

The command file SYEBLD.CMD contains instructions to the task builder to create the ...SYE task. There are several option statements in the command file that may be changed before task build to modify the standard analyzer task SYE:

1. The page width and buffer size for output from the terminal message processor to the terminals. These values are defined by the following TKB option statements:

   GBLDEF=PT$WTH:nnn (page width)

   and

   EXTSCT=TBUF:nnn (buffer size)

   where nnn is octal and must be the same in both directives. The default value is 110 (octal).

   The value assigned to nnn does not affect task size.

2. The page width and buffer size for the output from the error processors. These values are defined by the following TKB option statements:

   GBLDEF=PL$WTH:nnn (page width)

   and

   EXTSCT=OBUF:nnn (buffer size)

   where nnn is octal and must be the same in both directives. The default value is 120 (octal).

   Note that the value of nnn affects the size of the task.

Figure A-1 lists the contents of the command file SYEBLD.CMD.

```
;
;       [1,24]
;
;       S Y E B L D . C M D
;
;USE: RSX11M SYE ERROR LOG DISPLAY PROGRAM
;
;
;THIS COMMAND FILE WILL BUILD THE ...SYE TASK FROM THE
;SYEBLD.ODL FILE FOR A MAPPED SYSTEM
;
[1,54]SYE/MM/CP/-FP,MP:[1,34]SYE/CR/MA/-SP=IN:SYEBLD/MP
;
;
STACK=96
ASG=TI:1:5
TASK=...SYE
PAR=GEN:0:40000
;
;
;
;THIS DEFINES THE PAGE WIDTH FOR OUTPUT TO THE TERMINALS
;FROM THE TERMINAL MESSAGE PROCESSOR - THIS HAS NO EFFECT ON THE
;OUTPUT OF THE ERROR DISPLAYS
;ALL VALUES ARE IN OCTAL
;
GBLDEF=PT$WTH:110
;
```

Figure A-1 SYEBLD.CMD

```
;THIS IS THE SIZE OF THE TERMINAL BUFFER
;THIS VALUE MUST BE EQUAL TO THE VALUE USED FOR PT$WTH
;
EXTSCT=TBUF:110
;
;THIS DEFINES THE PAGE WIDTH FOR THE OUTPUT FROM THE ERROR
;PROCESSORS
;
;ALL VALUES ARE OCTAL
GBLDEF=PL$WTH:120
;
;THIS IS THE EXPANSION OF THE OUTPUT BUFFER
;THIS VALUE MUST BE THE SAME AS PL$WTH
;
EXTSCT=OBUF:120
;
;THIS IS THE DEFINITION OF THE PAGE LENGTH FOR THE OUTPUT FROM
;THE ERROR PROCESSORS
;
;ALL VALUES TYPED ARE OCTAL
;
GBLDEF=PL$LGH:74
;
/
```

Figure A-1 (Cont.) SYEBLD.CMD

Figure A-2 lists the contents of ERLBLD.CMD.

```
;
;       BOB LYONS APR 1, 76
;
;               BL001 -- ADDED "/" AT END OF OPTIONS
;
;               RA027 -- ADD GBLDEF FILE PROTECTION MASK
;                        FOR ERR.TMP&ERROR.TMP
;
[1,54]ERL/PR/MM,MP:[1,34]ERL/CR/MA/-SP=IN:ERLBLD/MP
UNITS=4
ASG=TI:2
ASG=SY:4
GBLDEF=FILPRO:6000
STACK=64
TASK=ERRRLOG
PAR=GEN:0:40000
UIC=[1,6]
PRI=145
/
```

Figure A-2 ERLBLD.CMD

APPENDIX B

ERR.TMP AND ERROR.TMP FILE RECORD FORMATS

This Appendix illustrates record formats that appear in a temporary file (either ERR.TMP or ERROR.TMP) created by the error logging task ERRLOG. Printouts generated by the RSX-11M/M-PLUS DMP utility program of either temporary file contain the record formats described in the following tables. Specifically, the tables describe:

- Initial configuration records (Table B-1),
- Device initialization records (Table B-2),
- Device hardware and interrupt timeout errors (Table B-3),
- Unexpected trap or interrupt errors (Table B-20),
- Device driver load and unload errors (Table B-21),
- Memory parity errors (Table B-22), and
- Power fail records (Table B-23).

Tables B-4 through B-18 explain the device-dependent parameters shown in Table B-3.

Users should refer to these tables when implementing new devices or features into the error logging system.

In printouts generated by the DMP program, the record size in bytes precedes each variable length record description.

Reserved Words - The term "reserved" means that the corresponding word contains a value for a specific purpose; its contents must not be altered.

Blank Words - The term "blank" means that the corresponding word was allocated for future use; its contents have no meaning.

Time Indications - Time indications are recorded as bytes or words. In general, byte entries are made by Executive routines; word entries are made by the get time system directive.

Table B-1
ERR.TMP and ERROR.TMP File Record Format
Initial Configuration Record

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	(Reserved) Error code (EC.INI)*	+2	E.CODE
2	Year	+4	E.TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	RSX-11M version number	+20	
9	Error logger Identification	+22	
10	Error logger Identification	+24	
11	Initial start (0) 'PSE' request (2)	+26	E.WHY
12	Configuration code	+30	
13	(Blank)	+32	E.NAME
14	(Blank)	+34	
15	(Blank)	+36	
16	(Blank)	+40	

*EC.INI=40

Configuration Codes:

0 - Unmapped

2 - 18 bit addressing

4 - 22 bit addressing

Table B-2
ERR.TMP and ERROR.TMP File Record Format
Device Initialization Record

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	Init. SUB-CODE* Error code (EC.INI)	+2	E.CODE
2	UCB address	+4	
3	Device name in ASCII	+6	
4	Physical unit number System unit number	+10	
5	Vector/4 System controller number	+12	
6	Device CSR Address	+14	
7	IOABM mask	+16	
8	Device control words (CW1)	+20	
9	Device control words (CW2)	+22	
10	Device control words (CW3)	+24	
11	Device control words (CW4)	+26	
12	(Blank)	+30	
13	(Blank)	+32	
14	(Blank)	+34	

* 1 = Device is loaded for error logging (E.DDW).
2 = Device is unloaded for error logging (E.RDDW).

Table B-3
ERR.TMP and ERROR.TMP File Record Format
Device Hardware and Interrupt Timeout Errors

Word			Offset Value	Definition
0	Size of record in bytes		+0	E.SIZE
1	(Reserved)	Entry Code (FC.DVC*)	+2	E.CODE
2	Minute	Second	+4	E.TIME
3	Day	Hour	+6	
4	Year	Month	+10	
5	Error sequence number		+12	E.SEQ
6	Active bit map		+14	E.ABM
7	Maximum retries	Retries left	+16	E.RTRY
8	I/O in Queue	Task priority	+20	E.IOC(21)
9	Task name		+22	E.TASK
10	Task name		+24	
11	Base address of task		+26	E.PAR
12	Programmer group	Programmer code	+30	E.UIC
13	UCB address		+32	E.UCB
14	Function code		+34	E.FNC
15	I/O Packet Parameters		+36	
16	(device-dependent)*		+40	
17	(device-dependent)		+42	
18	(device-dependent)		+44	
19	(device-dependent)		+46	
20	(device-dependent)		+50	
21	(device-dependent)		+52	
22	Count of registers		+54	E.RCNT
23	Device registers		+56	E.REGS
.	"	"	.	
.	"	"	.	
.	"	"	.	
n	Device registers		m	

*FC.DVC=1

* See I/O Packet Parameters in Tables B-4 through B-16.

Table B-4
I/O Packet Parameters
RK03, RK05, RK05F

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RKDA" for I/O GO	+46
20	LBN or transfer start	+50
21	(Reserved)	+52

Table B-5
I/O Packet Parameters
RP02, RP03

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RPCA" for I/O GO	+46
20	"RPDA" for I/O GO	+50
21	(Reserved)	+52

Table B-6
I/O Packet Parameters
RF11

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RFDAE" for I/O GO	+46
20	"RFDAR" for I/O GO	+50
21	(Reserved)	+52

Table B-7
I/O Packet Parameters
RS03, RS04

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	"RSDA" for I/O GO	+50
21	(Reserved)	+52

Table B-8
I/O Packet Parameters
RP04, RP05, RP06

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RPDC" for I/O GO	+46
20	"RPDA" for I/O GO	+50
21	(Reserved)	+52

Table B-9
I/O Packet Parameters
RK06

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RKDA" for I/O GO	+46
20	"RKDC" for I/O GO	+50
21	(Reserved)	+52

Table B-10
I/O Packet Parameters
TA11

Word		Offset Value
15	Space count if spacing function or "KTAPR" contents to transfer data through if data function	+36
16	Virtual buffer address of transfer	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	(Reserved)	+50
21	(Reserved)	+52

Table B-11
I/O Packet Parameters
TS03, TE10, TU10

Word		Offset Value
15	Space count if space function or transfer memory address (High) if data function	+36
16	Transfer memory address (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	(Reserved)	+50
21	(Reserved)	+52

Table B-12
I/O Packet Parameters
TS04

Word		Offset Value
15	Transfer memory address bits (high)	+16
16	Transfer memory address bits (low)	+20
17	Transfer size in bytes	+22
18	(Reserved)	+24
19	(Reserved)	+26
20	(Reserved)	+30
21	(Reserved)	+32

Table B-13
I/O Packet Parameters
TE16, TU16, TU45

Word		Offset Value
15	Space count if space function or transfer memory Address (High) if data function	+36
16	Transfer memory address (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	(Reserved)	+50
21	(Reserved)	+52

Table B-14
I/O Packet Parameters
TU58

Word		Offset Value
15	Relocation Bias of Data Buffer	+36
16	Buffer address of I/O Transfer	+40
17	Number of bytes to transfer	+42
18	Not used	+44
19	Low byte must be 0. High byte not used	+46
20	Logical or physical block number	+50
21	Not used	+52

Table B-15
I/O Packet Parameters
TC11

Word		Offset Value
15	Transfer memory address (High)	+36
16	Transfer memory address (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	LBN for transfer start	+50
21	(Reserved)	+52

Table B-16
I/O Packet Parameters
RL01, RL02

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Total transfer size in bytes	+42
10	Seek difference count	+44
11	"RLDA" for I/O GO	+46
12	This transfer size in bytes	+50
13	(Reserved)	+52

Table B-17
I/O Packet Parameters
RM03, RP07

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	(Reserved)	+44
11	"RMDC" for I/O GO	+46
12	"RMDA" for I/O GO	+50
13	(Reserved)	+52

Table B-18
I/O Packet Parameters
RX01

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	(Reserved)	+44
11	Current transfer size in bytes	+46
12	"RXDA" for I/O GO	+50
13	(Reserved)	+52

Table B-19
I/O Packet Parameters
RX02

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	Status register after interrupt	+44
11	Logical or physical block number	+46
12	Current transfer size in bytes	+50
13	Track number and sector number	+52

Table B-20
ERR.TMP and ERROR.TMP File Record Format
Unexpected Trap or Interrupt Error

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	(Reserved) Error Code (EC.NSI)	+2	E.CODE
2	Minute Second	+4	E.TIME
3	Day Hour	+6	
4	Year Month	+10	
5	Error Sequence Number	+12	E.SEQ
6	Active bit map	+14	E.ABM
7	Missed count Vector/4 of Trap	+16	E.VCTR(16) E.LOST(17)
8	Processor Status before trap	+20	
9	Program Counter before trap	+22	E.OPC

Table B-21
ERR.TMP and ERROR.TMP File Record Format
Device Driver Load or Unload

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	(200) Error code (EC.INI)	+2	E.CODE
2	Year	+4	E.TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	RSX-11M version number	+20	
9	Error logger Identification	+22	
10	Error logger Identification	+24	
11	Load(4)/Unload (10) code	+26	E.WHY
12	Configuration code	+30	
13	Driver name	+32	E.NAME
14	(Blank)	+34	
15	(Blank)	+36	
16	(Blank)	+40	

Table B-22
ERR.TMP and ERROR.TMP File Record Format
Memory Parity Errors

Word			Offset Value
0	Size		+0
1	Processor Type	Entry Code (EC.MPG*)	+2
2	MINUTE	SECOND	+4
3	DAY	HOUR	+6
4	YEAR	MONTH	+10
5	ERROR SEQUENCE NUMBER		+12
6	TRAP PC		+14
7	TRAP PS		+16
8	FIRST WORD TASK NAME		+20
9	SECOND WORD TASK NAME		+22
10	FIRST WORD PARTITION NAME		+24
11	SECOND WORD PARTITION NAME		+26
12	PARTITION BASE ADDRESS		+30
13	PARTITION SIZE		+32
14	MEMORY PARITY CSR's		+34
15	"		+36
16	"		+40
17	"		+42
18	"		+44
19	"		+46
20	"		+50
21	"		+52
22	"		+54
23	"		+56
24	"		+60
25	"		+62
26	"		+64
27	"		+66
28	"		+70
29	"		+72
30	Cache parity CSR's		+74
31	"		+100
32	"		+102
33	"		+104
34	"		+106
35	"		+110

*EC.MPG=2

Table B-23
ERR.TMP and ERROR.TMP File Record Format
Power Fail Record

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	Reserved Error Code (EC.PWR)	+2	E.CODE
2	Year	+4	E.TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	

EC.PWR=42

APPENDIX C

ERROR.SYS FILE RECORD FORMATS

This Appendix defines record formats contained in the file ERROR.SYS. If the report generating task SYE is not available, the user can consult these record formats to analyze the contents of ERROR.SYS. These formats should also be studied when adding features to the existing error logging system.

A user can run the system utility DMP to obtain a listing of ERROR.SYS. (See the RSX-11 Utilities Manual.) In such a printout, the record size in bytes precedes each variable length record.

Each record consists of one or more fields. The first field in every record is a header field that contains basic error information and optionally points to additional fields. The optional fields can follow in any order. Tables D-1 through D-6 define the six header field formats available for different types of errors; these errors are:

- Hardware or interrupt timeout errors (Table C-1),
- Memory parity errors (Table C-2),
- Error logging startup errors (Table C-3),
- Error logging termination errors (Table C-4),
- Power fail records (Table C-5)
- Errors caused by PSE's removal of the ERR.TMP file (Table C-6), and
- Unexpected trap or interrupt errors (Table C-7).

The first 8 words of any header field always contain the same type of information.

Tables C-8, C-25, C-26, and C-27 define four fields that may or may not be part of the record, depending on the type of error described. (Tables C-9 through C-20 describe device-dependent parameters referenced in Table C-8.) The four additional fields are:

- A program field, format 0, (Table C-8),
- A program field, format 2 (Table C-25),
- A bus activity field (Table C-26), and
- A device register field (Table C-27).

Reserved Words - The term "reserved" means that the corresponding word contains a value for a specific purpose; its contents must not be altered.

Blank Words - The term "blank" means that the related word was allocated for future use; its contents have no meaning.

Table C-1
ERROR.SYS File Record Format
Header Field Hardware (001) or Interrupt Timeout (140) Errors
Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (001/140) format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	UDE (device type and class)	+22	A$DCDT(22),A$DTYP(23)
10	UDE (controller and unit number)	+24	A$CNTN(24),A$UNTN(25)
11	Device mnemonic (ASCII)	+26	A$DVNM
12	Unit logical number	+30	A$SYUN
13	Offset to register field	+32	A$DRF
14	Offset to program field	+34	A$PGF
15	(Reserved)	+36	
16	(Reserved)	+40	
17	Offset to bus activity field	+42	A$BAF
18	(Reserved)	+44	

Table C-2
ERROR.SYS File Record Format
Header Field Memory Parity Error (002) Format 1

Word		Offset Value	Definition
0	Size of header Field in Bytes	+0	A$SIZE
1	Entry type (002) Format (1)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	UDE (device class and type)	+22	A$DCDT(22),A$DTYP(23)
10	UDE (controller and unit number)	+24	A$CNTN(24),A$UNTN(25)
11	Offset to program field	+26	A$MPF1
12	Offset to register field	+30	A$MRF1

Table C-3
ERROR.SYS File Record Format
Header Field System Start Up (040) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (040) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	(Reserved)	+22	
10	(Reserved)	+24	
11	(Reserved)	+26	

Table C-4
ERROR.SYS File Record Format
Header Field Error Logging Terminated (041) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (041) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	

Table C-5
ERROR.SYS File Record Format
Power Fail (042) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (42) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+18	A$SEQN

Table C-6
ERROR.SYS File Record Format
Header Field Error File (ERR.TMP) Re-initialized (043) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (043) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN

Table C-7
ERROR.SYS File Record Format
Header Field Unexpected Trap or Interrupt (141) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (141) Format (0)	+2	A$FMT(2) ,A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	Vector of undefined interrupt	+22	A$VCTR
10	Unexpected interrupts not logged because of this log effort	+24	A$UICT
11	PSW of interrupted process	+26	A$OPS
12	PC of interrupted process	+30	A$OPC
13	Offset to bus activity field	+32	A$UIPT

Table C-8
ERROR.SYS File Record Format
Program Field Format 0 (Hardware or Interrupt Timeout Errors)

The Program Field contains information about the program that caused the error or that was active when the error occurred.

Word		Offset Value	Definition
0	Size of program field in bytes	+0	
1	Address mode* Format (0)	+2	P$FMT(2) ,P$ADMD(3)
2	Task name in RAD50	+4	P$TN
3	Task name in RAD50	+6	
4	Programmer group Programmer code	+10	P$UIC
5	Base program load address**	+12	P$TA0
6	Function code	+14	P$FC
7	I/O packet information***	+16	P$PARM
8	(Device-dependent)	+20	
9	(Device-dependent)	+22	
10	(Device-dependent)	+24	
11	(Device-dependent)	+26	
12	(Device-dependent)	+30	
13	(Device-dependent)	+32	
14	Maximum retries Retries left****	+34	P$RT0
15	I/O still in Queue	+36	P$RQ

* 0 = Unmapped; 2 = 18-bit Addressing; 4 = 22-bit Addressing.

** Real if unmapped; 1/64 Real if 18-bit or 22-bit Addressing.

*** See I/O Packet Parameters, Tables C-9 through C-21.

**** If word is negative, error was not recovered.

Table C-9
I/O Packet Parameters
RK03, RK05, RK05F

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RKDA" for I/O GO	+26
12	LBN or transfer start	+30
13	(Reserved)	+32

Table C-10
I/O Packet Parameters
RP02, RP03

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RPCA" for I/O GO	+26
12	"RPDA" for I/O GO	+30
13	(Reserved)	+32

Table C-11
I/O Packet Parameters
RF11

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RFDAE" for I/O GO	+26
12	"RFDAR" for I/O GO	+30
13	(Reserved)	+32

Table C-12
I/O Packet Parameters
RS03, RS04

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	"RSDA" for I/O GO	+30
13	(Reserved)	+32

Table C-13
I/O Packet Parameters
RP04, RP05, RP06

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RPDC" for I/O GO	+26
12	"RPDA" for I/O GO	+30
13	(Reserved)	+32

Table C-14
I/O Packet Parameters
RK06

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RKDA" for I/O GO	+26
12	"RKDC" for I/O GO	+30
13	(Reserved)	+32

Table C-15
I/O Packet Parameters
TA11

Word		Offset Value
7	Space count if spacing function or "KTAPR" contents to transfer data through if data function	+16
8	Virtual buffer address of transfer	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	(Reserved)	+30
13	(Reserved)	+32

Table C-16
I/O Packet Parameters
TS03, TE10, TU10

Word		Offset Value
7	Space count if space function or transfer memory address (High) if data function	+16
8	Transfer memory address (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	(Reserved)	+30
13	(Reserved)	+32

Table C-17
I/O Packet Parameters
TS04

Word		Offset Value
15	Transfer memory address bits (high)	+16
16	Transfer memory address bits (low)	+20
17	Transfer size in bytes	+22
18	(Reserved)	+24
19	(Reserved)	+26
20	(Reserved)	+30
21	(Reserved)	+32

Table C-18
I/O Packet Parameters
TE16, TU16, TU45

Word		Offset Value
7	Space count if space function or transfer memory address (High) if data function	+16
8	Transfer memory address (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	(Reserved)	+30
13	(Reserved)	+32

Table C-19
I/O Packet Parameters
TU58

Word		Offset Value
15	Relocation bias of data buffer	+36
16	Buffer address of I/O transfer	+40
17	Number of bytes to transfer	+42
18	Not used	+44
19	Low byte must be zero. High byte not used	+46
20	Logical or physical block number used	+50
21	Not used	+52

Table C-20
I/O Packet Parameters
TC11

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	LBN for transfer start	+30
13	(Reserved)	+32

Table C-21
I/O Packet Parameters
RL01, RL02

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Total transfer size in bytes	+22
10	Seek difference count	+24
11	"RLDA" for I/O GO	+26
12	This transfer size in bytes	+30
13	(Reserved)	+32

Table C-22
I/O Packet Parameters
RM03, RP07

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RMDC" for I/O GO	+26
12	"RMDA" for I/O GO	+30
13	(Reserved)	+32

Table C-23
I/O Packet Parameters
RX01

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	Current transfer size in bytes	+26
12	"RXDA" for I/O GO	+30
13	(Reserved)	+32

Table C-24
I/O Packet Parameters
RX02

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	Status register often interrupt	+24
11	Logical or physical block number	+26
12	Current transfer size in bytes	+30
13	Track number and sector number	+32

Table C-25
ERROR.SYS File Record Format
Program Field Format 2 (Memory Parity Errors)

The Program Field contains information about the program that caused the error or that was active when the error occurred.

Word		Offset Value	Definition
0	Size of program field in bytes	+0	
1	Address mode* Format (2)	+2	P$FMT(2), P$ADMD(3)
2	Task name in RAD50	+4	P$TN
3	Task name in RAD50	+6	
4	Base address of load **	+10	P$TA2
5	Task maximum size **	+12	P$TS
6	Partition name in RAD50	+14	P$PN
7	Partition name in RAD50	+16	
8	Base address of partition	+20	P$PS
9	Processor Status word	+22	P$PSW
10	Program Counter	+24	P$PC

* 0 = Unmapped; 2 = 18-bit addressing; 4 = 22-bit addressing.

** Real if unmapped; 1/64th real if 18-bit or 22-bit addressing.

Table C-26
ERROR.SYS File Record Format
Bus Activity Field Format 0

The bus activity field contains information on bus activity at the time of the error.

Word			Offset Value	Definition
0	Size of bus activity field in bytes		+0	
1	(Reserved)	Format (0)	+2	B$fmt
2	Active vector/4	Active vector/4	+4	B$VEC
.	" "	" "	.	
.	" "	" "	.	
.	" "	" "	.	
n	Active vector/4	Active vector/4	+m	

Table C-27
ERROR.SYS File Record Format
Device Register Field Format 0

The device register field contains the contents of the control and status registers of the device on which the error occurred.

Word		Offset Value	Definition
0	Size of register field in bytes	+0	
1	(Reserved) Format (0)	+2	R$FMT
2	Device registers	+4	R$REG
.	" "	.	
.	" "	.	
.	" "	.	
n	Device registers	+m	

APPENDIX D

DEVICE DRIVER MODIFICATION

This Appendix describes the modifications made to the mass storage device drivers to enable error logging, as well as the common Executive routines that are supplied for the error logging subsystem. All error logging code is incorporated in the Executive or device driver at system generation. In addition to building the necessary device data structures, the system generation procedures enable all error logging code by defining the following conditional assembly symbols:

Symbol	Errors
E$$DVC	Hardware errors (including interrupt timeouts)
E$$NSI	Undefined interrupts
E$$PER	Memory parity errors

D.1 DEVICE DRIVER MODIFICATIONS

A device driver that is modified to enable error logging performs the following:

- Immediately before starting a device data type function, it calls the Executive routine "$BMSET" to set the appropriate bit, thereby indicating that the device has a data function active on the UNIBUS.
- It defines the entry point for interrupt handling via the system macro INTSE$.
- It calls the Executive routine $DVCER if it discovers a loggable error while executing the interrupt handling routine. The routine $DVCER then logs the error.
- It calls the Executive routine $DTOER if it discovers a device timeout error. The routine $DTOER then logs the error.
- For each error, it performs the required number of retries, and records in R2 the maximum number of possible retries (high byte) and the number of possible retries not taken (low byte). It then calls the Executive routine $IODON.

Before attempting to modify a device driver, a programmer should study a driver that has already been adapted for error logging, such as the RK05 (DKDRV) device driver.

D.2 EXECUTIVE ROUTINES

D.2.1 $BMSET

This routine is found in the ERROR module.

Calling sequence:

```
     CALL $BMSET
```

Description:

```
;+
; **-$BMSET-SET A DRIVER'S BIT IN THE I/O ACTIVE BITMAP
;
; THIS COROUTINE RAISES PROCESSOR PRIORITY TO SEVEN AND
; SETS THE MASK IN THE SCB IN $IOABM,
; LETS THE CALLER START THE FUNCTION, AND THEN,
; ALLOWS INTERRUPTS.
;
; INPUTS:
;
;        R4=ADDRESS OF THE SCB
;
; OUTPUTS:
;
;        $IOABM IS MODIFIED
;-
```

D.2.2 $DVCER

THIS ROUTINE IS FOUND IN THE ERROR module.

Calling sequence:

```
     CALL $DVCER
```

Description:

```
;+
; **-$DVCER-DEVICE ERROR BIT SET
;
; THIS IS THE EMB FORMATTING ROUTINE WHEN DEVICE ERROR BIT
; ERRORS ARE RECOGNIZED BY THE DRIVER. ON THE
; FIRST OCCURRENCE OF AN ERROR, $DVCER ATTEMPTS TO LOG IT.
; IF ERRORS OCCUR ON RETRIES, THEY ARE NOT LOGGED.
;
; THE ERROR CODE "EC.DVC" IS PUSHED ON THE STACK,
; THE ERROR IN PROGRESS BIT IS SET IN THE SCB, THE LENGTH OF THE
; REQUIRED EMB IS CALCULATED AND "$ALEMB" IS CALLED. IF "$ALEMB" FAILS
; TO ALLOCATE A PACKET FOR ANY REASON, "$DVCER" EXITS
; AND THE POINTER IN THE SCB TO THE EMB IS CLEARED.
; ELSE, THE SAVED $IOABM IS COPIED FROM THE SCB TO THE EMB AND
; A POINTER TO THE EMB IS SAVED IN THE SCB. THE ERROR INFORMATION
; INCLUDING DEVICE REGISTERS IS PUT INTO THE EMB AND THE
; RETURN IS MADE.
;
; INPUTS:
;
```

```
;       R4=ADDRESS OF THE SCB
;
; OUTPUTS:
;
;       IF SUCCESSFUL, THE EMB IS FILLED, AND THE SCB
;       CONTAINS A POINTER TO IT. AN ERROR IN PROGRESS BIT
;       IS SET IN THE SCB. ELSE, THE OCCURRENCE OF
;       THE ERROR IS COUNTED ONLY.
;
; NOTE: ALL REGISTERS ARE PRESERVED
;-
```

D.2.3 $DTOER

THIS ROUTINE IS IN THE ERROR module.

Calling sequence:

```
     CALL $DTOER
```

Description:

```
;+
; **-$DTOER-DEVICE TIMEOUTS
;
; THIS IS THE EMB FORMATTING ROUTINE WHEN
; TIMEOUT ERRORS ARE RECOGNIZED BY THE DRIVER. ON THE
; FIRST OCCURRENCE OF AN ERROR, "$DTOER" ATTEMPTS TO LOG IT.
; IF ERRORS OCCUR ON RETRIES, THEY ARE NOT LOGGED.
;
; THE ERROR CODE "EC.DTO" IS PUSHED ON THE STACK,
; THE ERROR IN PROGRESS BIT IS SET IN THE SCB, THE LENGTH OF THE
; REQUIRED EMB IS CALCUALTED AND "$ALEMB" IS CALLED. IF "$ALEMB" FAILS
; TO ALLOCATE A PACKET FOR ANY REASON, "$DTOER" EXITS
; AND THE POINTER IN THE SCB TO THE EMB IS CLEARED.
; ELSE, THE SAVED "$IOABM" IS COPIED FROM THE SCB TO THE EMB AND
; A POINTER TO THE EMB IS SAVED IN THE SCB. THE ERROR INFORMATION
; INCLUDING DEVICE REGISTERS IS PUT INTO THE EMB AND THE
; RETURN IS MADE. THE CONTENTS OF THE CSR THAT IS SAVED IS UNCHANGED
; FROM THE TIME OF TIMEOUT. AFTER THE CSR IS SAVED, DEVICE INTERRUPTS
; ARE DISABLED AND CPU PRIORITY IS LOWERED TO PR0.
;
; INPUTS:
;
;       R2=ADDRESS OF THE CSR
;       R4=ADDRESS OF THE SCB
;
; OUTPUTS:
;
;       C=0 IF FUNCTION WAS NOT A USER-MODE DIAGNOSTIC FUNCTION
;               THE EMB IS FILLED AND THE SCB CONTAINS A POINTER
;               TO IT AND ERROR IN PROGRESS FLAG IS SET IN THE SCB.
;       C=1 IF FUNCTION WAS A USER-MODE DIAGNOSTIC FUNCTION.
;               IN THIS CASE ONLY INTERRUPT ENABLE IS CLEARED AND
;               THE PRIORITY IS LOWERED TO 0.
;
;
; NOTE: ALL REGISTERS ARE PRESERVED
;-
```

D.2.4 $IODON

Refer to the RSX-11M Guide to Writing an I/O Driver.

D.2.5 INTSE$

$INTSE is the interrupt save routine for device drivers that support error logging. To generate the required interrupt service code in the driver, the programmer issues a call to the system macro INTSE$. INTSE$ also generates the appropriate global symbols for interrupt entry points.

The format of the INTSE$ macro is:

```
INTSE$    xx,pri,#ctlrs [,pssave,ucbsave]
```

where:

xx — is the 2-character device mnemonic.

pri — is the priority of the device (the priority that would be used in a call to $INTSE).

#ctlrs — is the number of controllers the driver services.

pssave — is an optional argument specifying a variable in which to save the PS word. If omitted, a variable named TEMP is used.

ucbsave — is an optional argument specifying a vector in which to store the interrupting device's UCB address. If omitted, a vector named CNTBL is used.

Outputs: R4 — is the controller index.

R5 — is the UCB address.

Example:

```
INTSE$ PP, PR4,P$$P11
```

D.2.6 Additional Executive Routines

These routines exist within the Executive and are called by the above mentioned routines. This information is supplied to assist in the handling and understanding of the internal workings of the RSX-11M error logging sub-system.

D.2.6.1 **$ALEMB or $ALEB1** - This routine is found in the ERROR module.

Calling sequence:

```
CALL $ALEMB    ; allocate EMB and increment error sequence number
CALL $ALEB1    ; allocate EMB and do not increment error sequence
                 number
```

Description:

```
;+
; **-$ALEMB-ALLOCATE AN ERROR MESSAGE BLOCK
; **-$ALEB1-ALLOCATE AN ERROR MESSAGE BLOCK (ALTERNATE ENTRY)
;
; THIS ROUTINE IS CALLED BY ERROR SERVICING
; ROUTINES. IT COUNTS THE OCCURRENCE OF THE ERROR AND TRIES
; TO ALLOCATE A CORE BLOCK FROM THE POOL.
; IF IT IS SUCCESFUL, IT FILLS IN THE ERROR CODE, THE TIME AND THE
; ERROR SEQUENCE NUMBER. ELSE, IT RETURNS WITH B-SET.
;
; INPUTS:
;
;       2(SP)=ERROR CODE
;       0(SP)=RETURN
;         R1 =SIZE OF THE EMB TO ALLOCATE
;
; OUTPUTS:
;
;       IF B-CLEAR:
;       R0=ADDRESS OF THE FIRST UNFILLED BYTE
;       R1=ADDRESS OF THE EMB
;
;       IF B-SET, UNSUCCESSFUL
;
; NOTE: R2 AND R3 ARE DESTROYED BY $ALEMB THRU $ALOCB
;-
```

D.2.6.2 **$QEMB** - This routine is found in the ERROR module.

Calling sequence:

```
     CALL $QEMB
```

Description:

```
;+
; **-$QEMB-QUEUE AN EMB
;
; THIS IS THE COMMON POINT FOR ALL EMBS. THE EMB IS QUEUED
; FIFO IN THE ERROR QUEUE.  IF THERE ARE
; ENOUGH BYTES OF EMBS IN THE POOL, THE LOGGER TASK IS AWAKENED.
; ELSE, IF THE QUEUE WAS EMPTY, A SCHEDULE REQUEST IS MADE
; SO A QUEUED EMB WILL BE WRITTEN WITHIN A TIME LIMIT.
; ELSE, A RETURN IS MADE.
;
; INPUTS:
;
;       R1=ADDRESS OF THE EMB
;
; OUTPUTS:
;
;       THE EMB IS QUEUED. CONDITIONALLY, THE LOGGER IS WAKED
;       OR A SCHEDULE REQUEST IS MADE FOR THE LOGGER
;
; NOTE: REGISTERS R0-R3 ARE DESTROYED
;-
```

A

B

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

RSX-11M/M-PLUS
Error Logging
Reference Manual
AA-H270A-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify)_____

Name_____ Date_____

Organization_____

Street_____

City_____ State_____ Zip Code or Country_____

Please cut along this line.

Do Not Tear - Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

RT/C SOFTWARE PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

Do Not Tear - Fold Here

Cut Along Dotted Line